



# **Amazon Web Services Data Engineering Immersion Day**

---

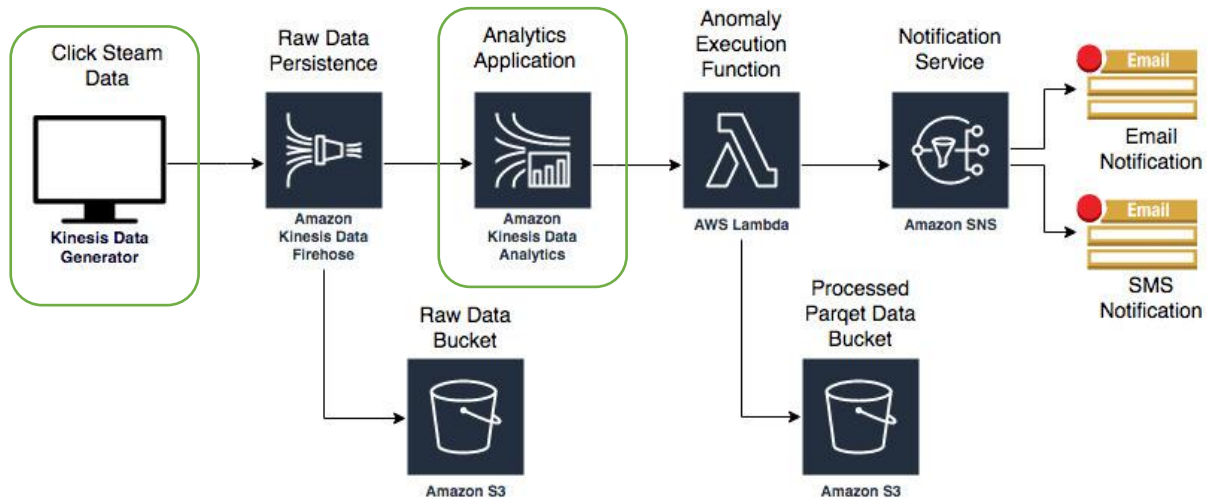
Lab 3 - Prelab. Real-Time Clickstream Anomaly Detection

## Table of Contents

<b><i>Introduction .....</i></b>	<b><i>2</i></b>
<b><i>Get Started Using the Lab Environment .....</i></b>	<b><i>3</i></b>
<b><i>CloudFormation Stack Deployment .....</i></b>	<b><i>5</i></b>
<b><i>Set up the Amazon Kinesis Data Generator .....</i></b>	<b><i>8</i></b>
<b><i>Set up Email and SMS Subscription .....</i></b>	<b><i>10</i></b>
<b><i>Review AWS Lambda Anomaly function:.....</i></b>	<b><i>11</i></b>

## Introduction

This guide will help you set up the pre-lab environment for the Real-Time Clickstream Anomaly Detection Amazon Kinesis Data Analytics lab.



After you deploy the CloudFormation template, sign into your account to view the following resources:

- Two Amazon Simple Storage Service (Amazon S3) buckets: You will use these buckets to persist raw and processed data.
- One AWS Lambda function: This Lambda function will be triggered once an anomaly has been detected.
- Amazon Simple Notification Service (Amazon SNS) topic with an email and phone number subscribed to it: The Lambda function will publish to this topic once an anomaly has been detected.
- Amazon Cognito User credentials: You will use these user credentials to log into the Kinesis Data Generator to send records to our Amazon Kinesis Data Firehose.

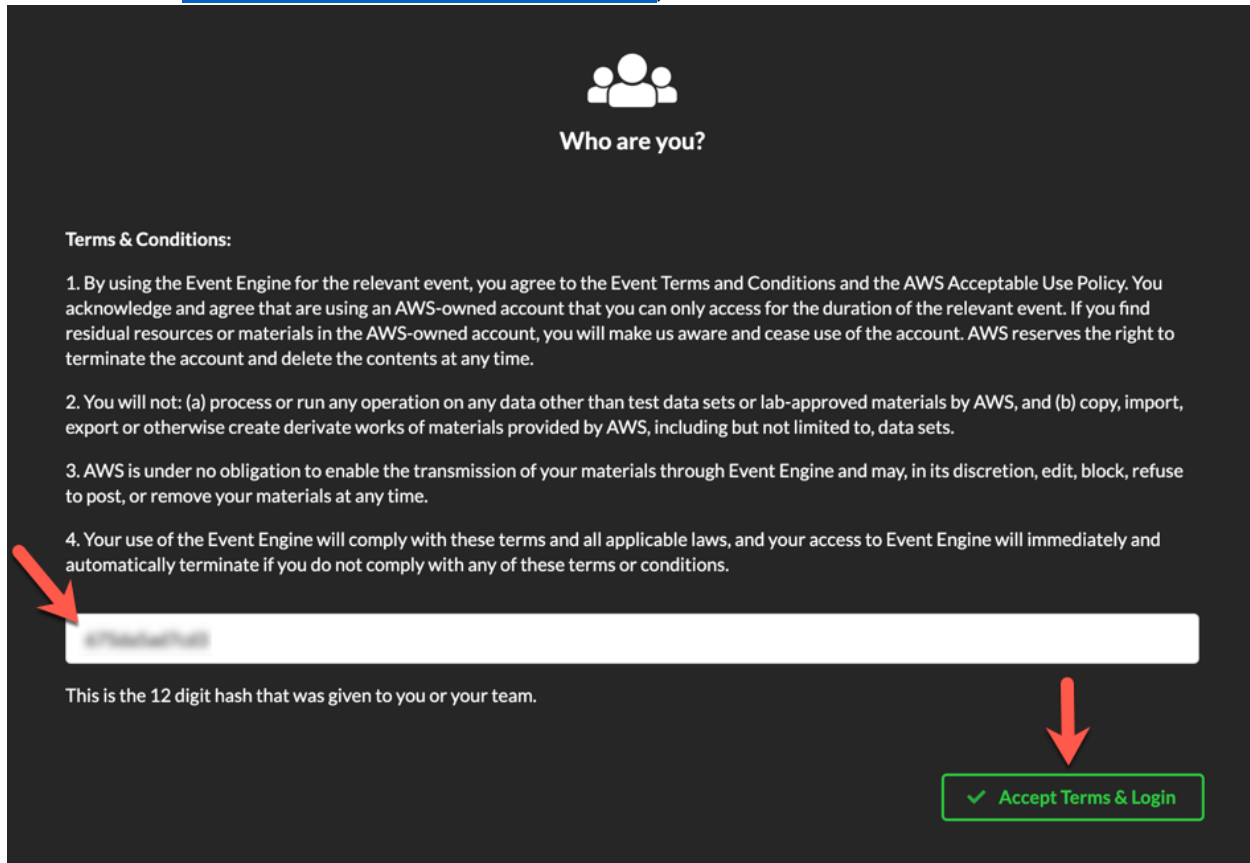
Today, you are attending a formal AWS event and we've uploaded the CloudFormation template and the Kinesis Data Generator to an s3 bucket for you. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - <https://github.com/aws-samples/data-engineering-for-aws-immersion-day>.

## Get Started Using the Lab Environment

Today, you are attending a formal event and you will have been sent your access details beforehand. If in the future you might want to perform these labs in your own AWS environment by yourself, you can follow instructions on GitHub - <https://github.com/aws-samples/data-engineering-for-aws-immersion-day>.

A 12-character access code (or 'hash') is the access code that grants you permission to use a dedicated AWS account for the purposes of this workshop.

1. Go to <https://dashboard.eventengine.run/>, enter the access code and click Proceed:



Who are you?

**Terms & Conditions:**

1. By using the Event Engine for the relevant event, you agree to the Event Terms and Conditions and the AWS Acceptable Use Policy. You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through Event Engine and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of the Event Engine will comply with these terms and all applicable laws, and your access to Event Engine will immediately and automatically terminate if you do not comply with any of these terms or conditions.

This is the 12 digit hash that was given to you or your team.

✓ Accept Terms & Login

2. On the Team Dashboard web page you will see a set of connection strings and parameters that you will need during the labs. Best to save them to a text file locally, alternatively you can always go to this page to review them. Replace the parameters with the corresponding values from here where indicated in subsequent labs:

Because you're at a formal event, some AWS resources have been pre-deployed for your convenience, for example

- RDS Postgres database that you will use as your source endpoint (parameter **DMSInstanceEndpoint**)

## Lab 3 - Prelab. Real-Time Clickstream Anomaly Detection

### Modules

#### DMS\_Student\_Prereqs

Outputs:

##### Data Engineering Workshop

Parameter	Value
BucketName	mod-08b80667356c4f8a-dmslabs3bucket-1ijtekr232zk
BusinessAnalystUser	mod-08b80667356c4f8a-BusinessAnalystUser-1DPVYKJ8G0JK3
BusinessAnalystUserPolicy	BusinessAnalystUserPolicy
DMSLabRoles3	arn:aws:iam::433083714985:role/mod-08b80667356c4f8a-DMSLabRoles3-10V87K4LU3P66
GlueLabRole	mod-08b80667356c4f8a-GlueLabRole-HB1L2G7U4DU8
S3BucketWorkgroupA	mod-08b80667356c4f8a-s3bucketworkgroupa-1sw7181wwqp6o
S3BucketWorkgroupB	mod-08b80667356c4f8a-s3bucketworkgroupb-10cz7ir988eoh
WorkgroupManagerUser	mod-08b80667356c4f8a-WorkgroupManagerUser-1DSHJDROQWRMZ
WorkgroupManagerUserPolicy	WorkgroupManagerUserPolicy

#### DMS\_Instructor\_Prereqs

Outputs:

##### Data Source for DMS Lab

Parameter	Value
DMSInstanceEndpoint	dmslabinstance.ckyqv1sdkm8m.us-east-1.rds.amazonaws.com
CDCFunction	arn:aws:lambda:us-east-1:433083714985:function:GenerateCDCData

3. On the Team Dashboard, please click AWS Console to log into the AWS Management Console:

### Team Dashboard

Event

 AWS Console

 SSH Key

#### Event: Macquarie Bank Data Engineering Immersion Day - Test

Team Name: Igor Izotov

Event ID: d2302d4ae9ff4ea2857846b74f7de7e2

Team ID: 1c2f7ad7ec044b0b8276f917c5983133

4. Click Open Console. For the purposes of this workshop, you will not need to use command line and API access credentials

S

## CloudFormation Stack Deployment

1. Use this link to create a new CloudFormation Stack:

<https://console.aws.amazon.com/cloudformation/home?region=us-east-1#/stacks/new?stackName=kinesis-pre-lab&templateURL=https://dataeng-immersion-day.s3.amazonaws.com/lab3-us-east-1.json>

Create stack

**Prerequisite - Prepare template**

Prepare template  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source  
Selecting a template generates an Amazon S3 URL where it will be stored.

☒ Amazon S3 URL ☐ Upload a template file

Amazon S3 URL  
  
Amazon S3 template URL

S3 URL: <https://dataeng-immersion-day.s3.amazonaws.com/lab3-us-east-1.json>

2. Click **Next** at the bottom of the page in as shown in above screenshot.
3. In the **Parameters** section, fill the following fields as shown in screenshot:
  - **Username:** This is your username to login to the Kinesis Data Generator
  - **Password:** This is your password for the Kinesis Data Generator. The password must be at least 6 alpha-numeric characters and contain at least one number and a capital letter.
  - **Email:** Type an email address that you can access. The SNS topic sends a confirmation to this address.
  - **SMS:** Type a phone number (+1XXXXXXXXXX) where you can receive texts from the SNS topic.
  - **CodeS3Bucket:** Since you are at a formal AWS event, please accept the default value here.

## Lab 3 - Prelab. Real-Time Clickstream Anomaly Detection

CloudFormation > Stacks > kin-lab > Update stack

Step 1  
Specify template

Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review

### Specify stack details

**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**Kinesis Pre Lab set up**

**Username**  
The username of the user you want to create in Amazon Cognito.

test

**Password**  
The password of the user you want to create in Amazon Cognito. Must be at least 6 alpha-numeric characters, and contain at least one number

test1234

**email**  
Email address to send anomaly detection events.

**SMS**  
Mobile Phone number to send SMS anomaly detection events. +1XXXXXXXXXX

**Other parameters**

**CodeS3bucket**  
Please enter bucket name where you have uploaded datagen-cognito-setup.zip

Cancel Previous Next

4. In the **Options**, section, keep the default values.
5. In the **Review** section, select the check box marked **I acknowledge that AWS CloudFormation might create IAM resources**.

Capabilities

**The following resource(s) require capabilities: [AWS::IAM::Role]**

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions.  
[Learn more](#)

☒ I acknowledge that AWS CloudFormation might create IAM resources.

Cancel Previous Create change set Create stack

## Lab 3 - Prelab. Real-Time Clickstream Anomaly Detection

- Click **Create**. CloudFormation redirects you to your existing stacks. The **kinesis-pre-lab** displays a **CREATE\_IN\_PROGRESS** status.

CloudFormation > Stacks > Kinesis-Pre-Lab

**Stacks (13)**

Filter by stack name

Active

View nested

Kinesis-Pre-Lab  
2020-01-30 02:28:54 UTC-0800  
CREATE\_COMPLETE

aws-cloud9-termianl-2ab1a458585b4f80b5366de520327720

Create Stack Actions Design template

Filter: Active By Stack Name Showing 20 stacks

Stack Name	Created Time	Status	Description
kinesis-pre-lab	2018-07-31 14:15:26 UTC-0400	CREATE_IN_PROGRESS	Supporting elements for the Kinesis Analytics click stream lab

- Once your stack is deployed, click the **Outputs** tab to view more information:
  - KinesisDataGeneratorUrl**: This value is the Kinesis Data Generator (KDG) URL.
  - RawBucketName** – Store raw data coming from KDG.
  - ProcessedBucketName** – Store transformed data

Stacks kinesis-pre-lab Delete Update Stack actions Create stack

Stack info Events Resources **Outputs** Parameters Template Change sets

**Outputs (3)**

Search outputs

Key	Value	Description	Export name
KinesisDataGeneratorUrl	<a href="https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-east-1_SCwIY35nT&amp;ipid=us-east-1:9dc3be32-cd44-47bd-8d1c-3b58b2be48b2&amp;cid=2jagps6k0dm939d67st9gl2s0f&amp;r=us-east-1">https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-east-1_SCwIY35nT&amp;ipid=us-east-1:9dc3be32-cd44-47bd-8d1c-3b58b2be48b2&amp;cid=2jagps6k0dm939d67st9gl2s0f&amp;r=us-east-1</a>	The URL for your Kinesis Data Generator.	-
ProcessedBucketName	kinesis-pre-lab-processed3bucket-m703w7hqxm6	This the bucket name of where your Processed data will be store at	-
RawBucketName	kinesis-pre-lab-raws3bucket-1aa7n8xmrga4	This the bucket name of where your Raw data will be store at	-

Congratulations! You are all done with the CloudFormation deployment.



## Set up the Amazon Kinesis Data Generator

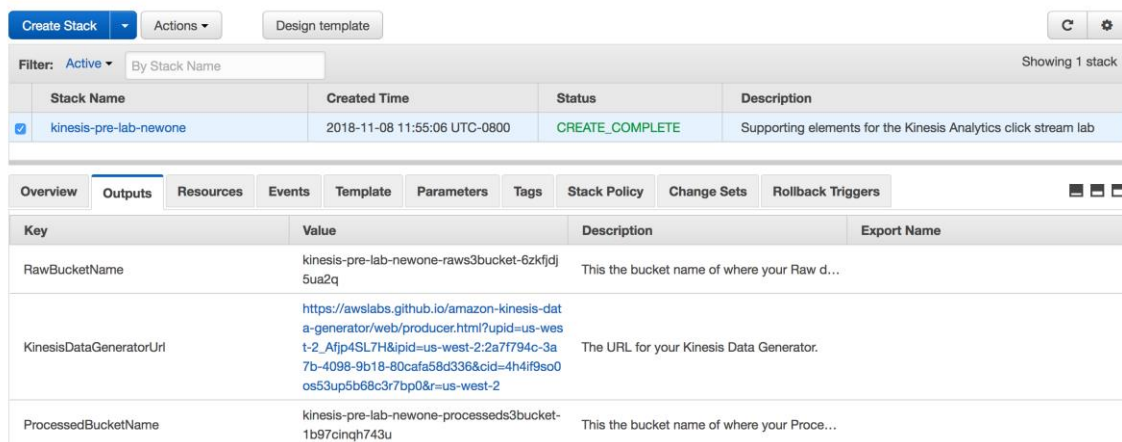
On the **Outputs** tab, notice the **Kinesis Data Generator URL**. Navigate to this URL to login into the Amazon Kinesis Data Generator (Amazon KDG).

The KDG simplifies the task of generating data and sending it to Amazon Kinesis. The tool provides a user-friendly UI that runs directly in your browser. With the KDG, you can do the following tasks:

- Create templates that represent records for your specific use cases
- Populate the templates with fixed data or random data
- Save the templates for future use
- Continuously send thousands of records per second to your Amazon Kinesis stream or Firehose delivery stream

Let's test your Cognito user in the Kinesis Data Generator.

1. On the **Outputs** tab, click the **KinesisDataGeneratorUrl**.

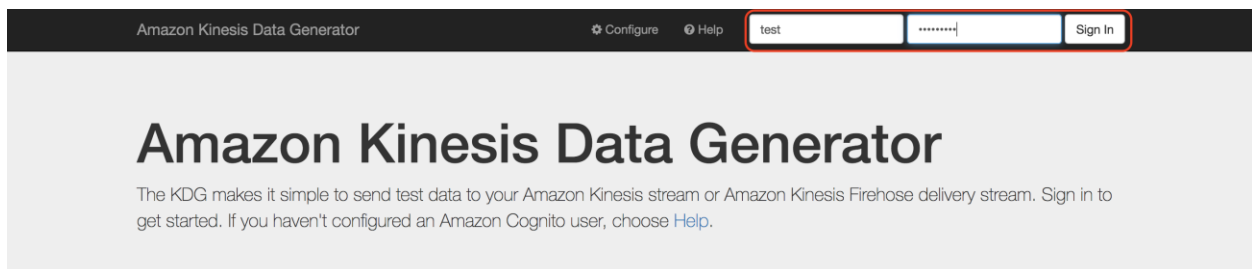


Stack Name	Created Time	Status	Description
kinesis-pre-lab-newone	2018-11-08 11:55:06 UTC-0800	CREATE_COMPLETE	Supporting elements for the Kinesis Analytics click stream lab

Key	Value	Description	Export Name
RawBucketName	kinesis-pre-lab-newone-raws3bucket-6zkfjdj5ua2q	This the bucket name of where your Raw d...	
KinesisDataGeneratorUrl	<a href="https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-west-2_Afjp4SL7H&amp;ipld=us-west-2:2a7f794c-3a7b-4098-9b18-80cfa58d336&amp;cid=4h4if9so0os53up5b68c3r7bp0&amp;r=us-west-2">https://awslabs.github.io/amazon-kinesis-data-generator/web/producer.html?upid=us-west-2_Afjp4SL7H&amp;ipld=us-west-2:2a7f794c-3a7b-4098-9b18-80cfa58d336&amp;cid=4h4if9so0os53up5b68c3r7bp0&amp;r=us-west-2</a>	The URL for your Kinesis Data Generator.	
ProcessedBucketName	kinesis-pre-lab-newone-processed3bucket-1b97cinqh743u	This the bucket name of where your Proce...	

2. Sign in using the **username** and **password** you entered in the CloudFormation console.



3. After you sign in, you should see the KDG console. You need to set up some templates to mimic the clickstream web payload.

### Lab 3 - Prelab. Real-Time Clickstream Anomaly Detection

- Create the following three templates. Copy the tab name highlight in bold letter and value as json string, refer screenshot:

#### Schema Discovery Payload

```
{"browseraction": "DiscoveryKinesisTest", "site": "yourwebsiteurl.domain.com"}
```

#### Click Payload

```
{"browseraction": "Click", "site": "yourwebsiteurl.domain.com"}
```

#### Impression Payload

```
{"browseraction": "Impression", "site": "yourwebsiteurl.domain.com"}
```

- Change Region to **US-EAST-1** and select a created Firehose Delivery Stream from the dropdown.
- Set **Records per second** to **1**

Your Amazon Kinesis Data Generator console should look similar to this example.

The screenshot shows the Amazon Kinesis Data Generator console interface. At the top, there's a header bar with "Amazon Kinesis Data Generator" and a settings icon. Below this, the configuration is organized into sections:

- Region:** A dropdown menu set to "us-east-1".
- Stream/delivery stream:** A dropdown menu showing "Kinesis-Pre-Lab-FirehoseDeliveryStream-1XMH0FAX1".
- Records per second:** Two tabs, "Constant" and "Periodic", with "Periodic" selected. Below the tabs is a text input field containing the value "1".
- Compress Records:** A checkbox that is currently unchecked.
- Record template:** A section with five tabs: "Schema Discovery Payload" (which is highlighted in bold), "Click Payload", "Impression Payload", "Template 4", and "Template 5". Below the tabs, there's a text area displaying the JSON payload for the selected template: 

```
{"browseraction": "DiscoveryKinesisTest", "site": "yourwebsiteurl.domain.com"}
```

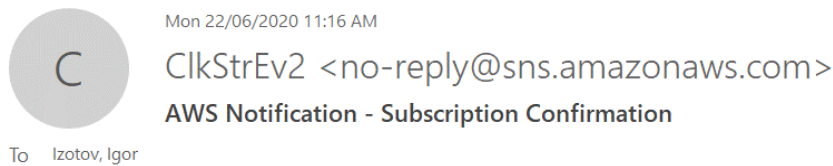
**Don't click on Send Data yet, leave this browser tab open, we will do that during the main lab.**

## Set up Email and SMS Subscription

1. In a new browser tab, go to Amazon SNS Topics: by following this link:  
<https://console.aws.amazon.com/sns/v3/home?region=us-east-1#/topics>
2. Click the topic name. The Topic details screen appears listing the e-mail/SMS subscription as pending or confirmed.

The screenshot shows the Amazon SNS console interface. On the left is a navigation menu with options like Dashboard, Topics, Subscriptions, Mobile, Push notifications, and Text messaging (SMS). The main panel displays the 'ClickStreamEvent' topic details. It includes fields for Name (ClickStreamEvent), ARN (arn:aws:sns:us-east-1:722911934590:ClickStreamEvent), Display name (ClickStreamEvent), and Topic owner (redacted). Below this are tabs for Subscriptions, Access policy, Delivery retry policy (HTTP/S), Delivery status logging, Encryption, and Tags. The 'Subscriptions (2)' tab is active, showing a table with two subscriptions. The first subscription has ID 13ed5fe6-6caa-49f4-8efc-af801997879a, Endpoint [redacted].com, Status Confirmed, and Protocol EMAIL. The second subscription has ID 276d59b2-b91c-4f83-ab9f-7cdf3acc6f8b, Endpoint +1 [redacted], Status Confirmed, and Protocol SMS.

3. Check your inbox for a subscription confirmation email from [no-reply@sns.amazonaws.com](mailto:no-reply@sns.amazonaws.com), click **Confirm subscription** to confirm



You have chosen to subscribe to the topic:  
**arn:aws:sns:us-east-1:222752441477:ClickStreamEvent2**

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):  
[Confirm subscription](#)

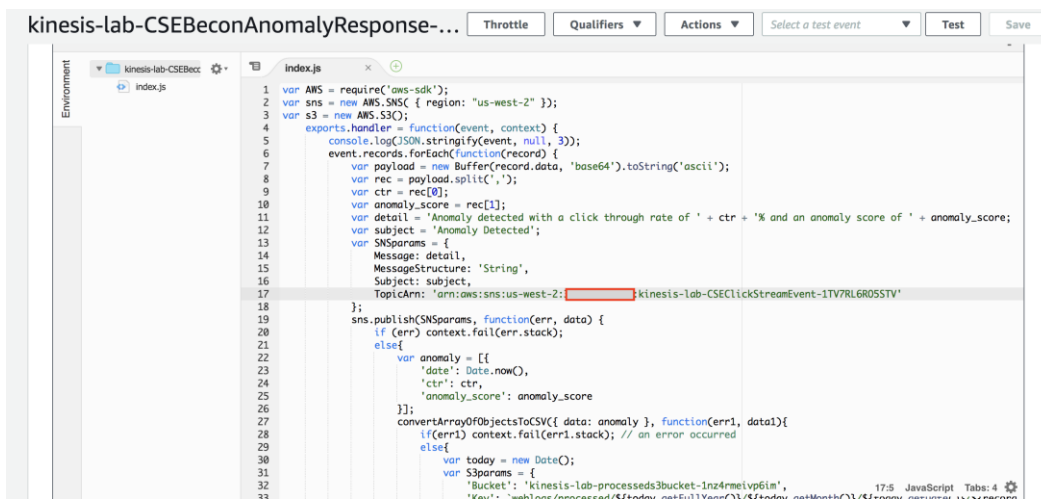
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

**Note:** If you can't locate the request confirmation email, make sure to check your email junk folder.

### Review AWS Lambda Anomaly function:

CloudFormation template already deployed this Lambda function. You just need to spend few minutes to observe code and understand the action behind the lambda trigger:

1. In the console, navigate to **CSEBeconAnomalyResponse** AWS Lambda function by following the link: <https://console.aws.amazon.com/lambda/home?region=us-east-1#/functions/CSEBeconAnomalyResponse?tab=configuration>
2. Scroll down to code section.



```
1 var AWS = require('aws-sdk');
2 var sns = new AWS.SNS({ region: 'us-west-2' });
3 var s3 = new AWS.S3();
4 exports.handler = function(event, context) {
5   console.log(JSON.stringify(event, null, 3));
6   event.records.forEach(function(record) {
7     var payload = new Buffer(record.data, 'base64').toString('ascii');
8     var rec = payload.split(',');
9     var ctr = rec[0];
10    var anomaly_score = rec[1];
11    var detail = 'Anomaly detected with a click through rate of ' + ctr + '% and an anomaly score of ' + anomaly_score;
12    var subject = 'Anomaly Detected';
13    var SNSparams = {
14      Message: detail,
15      MessageStructure: 'String',
16      Subject: subject,
17      TopicArn: 'arn:aws:sns:us-west-2:kinesis-lab-CSEClickStreamEvent-1TV7RL6R05STV'
18    };
19    sns.publish(SNSparams, function(err, data) {
20      if (err) context.fail(err.stack);
21    } else {
22      var anomaly = [{
23        'date': Date.now(),
24        'ctr': ctr,
25        'anomaly_score': anomaly_score
26      }];
27      convertArrayOfObjectsToCSV({ data: anomaly }, function(err1, data1){
28        if(err1) context.fail(err1.stack); // an error occurred
29      } else {
30        var today = new Date();
31        var S3params = {
32          'Bucket': 'kinesis-lab-processed3bucket-1nz4rme1vp6im',
33          'Key': 'anomaly/processed/ETtoday-' + today.getFullYear() + '/' + today.getMonth() + '/' + today.getDate() + '/' + today.getHours() + '/' + today.getMinutes() + '/' + today.getSeconds() + '.csv'
34        };
35        s3.putObject(S3params, function(err2, data2) {
36          if(err2) context.fail(err2.stack);
37        } else {
38          console.log('Anomaly detected and stored in S3');
39        }
40      }
41    }
42  });
43 }
```

3. Review the code in the Lambda code editor. Notice the TopicArn value matches the SNS topic ARN from the previous step.

**You've completed the pre-lab instructions. Please proceed to lab 3.**